

# Marvel

Diego Rodríguez Riera

21 de noviembre de 2018

## Índice

<b>1. Definición del problema</b>	<b>3</b>
<b>2. Análisis del problema</b>	<b>3</b>
<b>3. Modelo Entidad-Relación</b>	<b>4</b>
<b>4. Modelo Relacional</b>	<b>5</b>
<b>5. Restricciones</b>	<b>6</b>
5.1. HabilityIsAvailable . . . . .	6

## 1. Definición del problema

Unos frikis de Marvel desean estudiar el comportamiento de los superhéroes para poder predecir el resultado de los combates entre ellos. Para ello, almacenarán para los superhéroes y supervillanos información acerca de su nombre, los poderes que disponen y el nivel de ese poder en un intervalo de 0 - 100.

Cada vez que hay un combate entre superhéroes y supervillanos desean almacenar el efecto que produce en cada uno de ellos los lances. Así, para cada lance desean conocer cuándo se produce, entre qué par de actores y el efecto sobre los diferentes poderes que tienen esos actores.

Por otra parte, los supervillanos y superhéroes pueden recuperar poderes con el tiempo (si no están en una lucha), por lo que es de interés conocer cada cuanto tiempo, la cantidad o porcentaje de poder que pueden recuperar, y para cada poder tienen un límite mínimo que una vez sobrepasado pierden el combate.

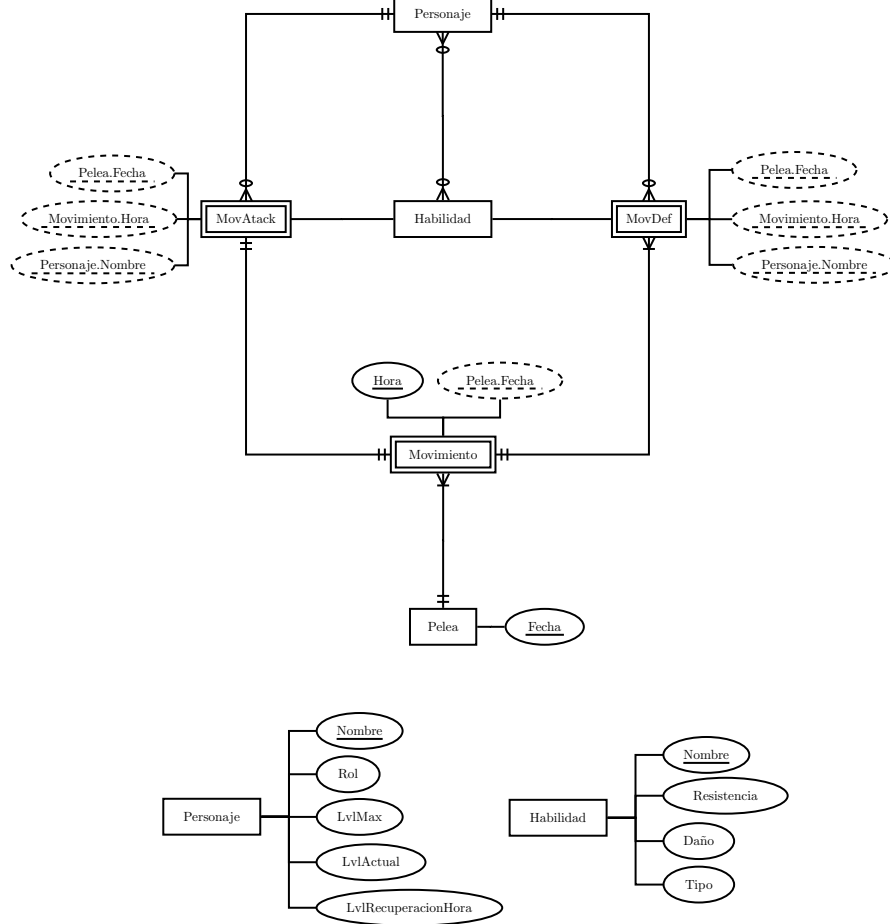
Diseñar el modelo relacional de la base de datos.

## 2. Análisis del problema

Necesitamos almacenar los diferentes movimientos de varios personajes, los cuales forman una "pelea", estos son los personajes de nuestro juego, estos tienen un rol sea villano o heroe, también tienen un nombre, un nivel máximo de poder, un nivel de poder actual, una serie de habilidades con un nivel, y un nivel de poder o vida actual.

Por otra parte, tenemos los enfrentamientos, que están formados por una serie de movimientos, en los que se ven involucrados como mínimo un actor y un receptor del ataque o habilidad, el cual tendrá una penalización.

### 3. Modelo Entidad-Relación



## 4. Modelo Relacional

**Personaje** (Nombre, Rol, Habilidad, LvlMax, LvlActual,  
LvlRecuperacionHora)

**Movimiento** (Hora, Fecha\*)  
FK(Fecha=Pelea.Fecha)

**MovAtack** (Hora\*, Fecha\*, Personaje\*, Habilidad\*)  
FK(Hora=Movimiento.Hora)  
FK(Fecha=Pelea.Fecha)  
FK(Personaje=Personaje.Nombre)  
FK(Habilidad=Habilidad.Nombre)  
trigger(HabilityIsAvailable)

**MovDef** (Hora\*, Fecha\*, Personaje\*, Habilidad\*)  
FK(Hora=Movimiento.Hora)  
FK(Fecha=Pelea.Fecha)  
FK(Personaje=Personaje.Nombre)  
FK(Habilidad=Habilidad.Nombre)  
trigger(HabilityIsAvailable)

**Pelea** (Fecha)

**Habilidad** (Nombre, Resistencia, Daño, Tipo)

**Personage-Habilidad** (Personaje\*, Habilidad\*)  
FK(Personaje=Personaje.Nombre)  
FK(Habilidad=Habilidad.Nombre)

## 5. Restricciones

Sentencias sql las cuales garantizaran la consistencia de la base de datos

### 5.1. HabilityIsAvailable

```
create trigger HabilityIsAvailable1
instead of insert on MovAttack
begin
    if (Habilidad in
        select ph.Habilidad
        from Personaje-Hablilidad as ph
        where ph.Personaje = Personaje
    ) then
        insert into MovAttack
        values (Hora, Fecha, Personaje, Habilidad);
end;

create trigger HabilityIsAvailable2
instead of insert on MovDef
begin
    if (Habilidad in
        select ph.Habilidad
        from Personaje-Hablilidad as ph
        where ph.Personaje = Personaje
    ) then
        insert into MovDef
        values (Hora, Fecha, Personaje, Habilidad);
end;
```